STA 445 Final Exam

Noah Plant

December 11, 2024

Exam Questions

Question 1 [20 points]

I am interested in the average attendance at each World Cup dependent on the host country.

a. Scrape this information from the Wikipedia page: 'https://en.wikipedia.org/wiki/FIFA_World_Cup'. Provide the code for obtaining the proper table from the Wikipedia page.

```
# Load in the html pages, This is done in a seperate chunk so I do not have to
# continously load in the page.

url<-'https://en.wikipedia.org/wiki/FIFA_World_Cup'

page<-read_html(url)

myTable<-page%>%html_nodes('table')%>%
  .[[4]]%>%
  html_table(header=FALSE,fill=TRUE)
```

b. Clean the data you have scraped to include the following columns: `Year`, `Hosts`, `Matches`, `Totalattendance`, and `Averageattendance`. Assign the data.frame to the object `World_Cup`. You will either need to make your own column names or properly clean the strings given for the column names (they contain special characters that should not be retained). Remove commas from numerical values and ensure the Attendance columns are properly formatted as numerical data. Keep the `Year` variable as strings or factors. Remove data related to any World Cups that have not occurred and the Overall statistics. Show the `head()` of `World_Cup` when finished.

```
# Select the correct data
World_Cup<-myTable%>%slice(-28,-27,-26,-25)%>%
  slice(-1,-2)%>%
  select(c(1,2,4,5,6))

# Rename the columns
colnames(World_Cup)<-c("Year","Hosts","Totalattendance","Matches","Averageattendance")

# Clean the data inside
World_Cup<-World_Cup%>%
  mutate(Totalattendance=str_replace_all(Totalattendance,pattern=',',replacement=''))%>%
  mutate(Averageattendance=str_replace_all(Averageattendance,pattern=",",replacement=''))%>%
  mutate(Totalattendance=as.numeric(Totalattendance))%>%
```

```
  mutate(Averageattendance=as.numeric(Averageattendance))%>%
  mutate(Matches=as.numeric(Matches))

# Show off the data

head(World_Cup)
```

```
## # A tibble: 6 x 5
##   Year  Hosts       Totalattendance Matches Averageattendance
##   <chr> <chr>                 <dbl>   <dbl>             <dbl>
## 1 1930  Uruguay              590549      18             32808
## 2 1934  Italy                363000      17             21353
## 3 1938  France               375700      18             20872
## 4 1950  Brazil              1045246      22             47511
## 5 1954  Switzerland          768607      26             29562
## 6 1958  Sweden               819810      35             23423
```

c. Some countries have hosted multiple World Cups. Make unique identifiers for each World Cup by pasting together the Host and Year. Create a new column named `WorldCup` that contains these unique identifiers (i.e. `Uruguay1930`). Remove any remaining spaces in the `WorldCup` names. Remove the `Hosts` and `Year` columns when finished.

```
World_Cup<-World_Cup%>%mutate(WorldCup=str_c(Hosts,Year,sep=""))%>%
  mutate(WorldCup=str_replace_all(WorldCup,regex("\\s*"),""))

World_Cup<-World_Cup%>%select(-Hosts,-Year)
```

d. Display the head of the data frame `World_Cup`.

```
head(World_Cup)
```

```
## # A tibble: 6 x 4
##   Totalattendance Matches Averageattendance WorldCup
##             <dbl>   <dbl>             <dbl> <chr>
## 1          590549      18             32808 Uruguay1930
## 2          363000      17             21353 Italy1934
## 3          375700      18             20872 France1938
## 4         1045246      22             47511 Brazil1950
## 5          768607      26             29562 Switzerland1954
## 6          819810      35             23423 Sweden1958
```

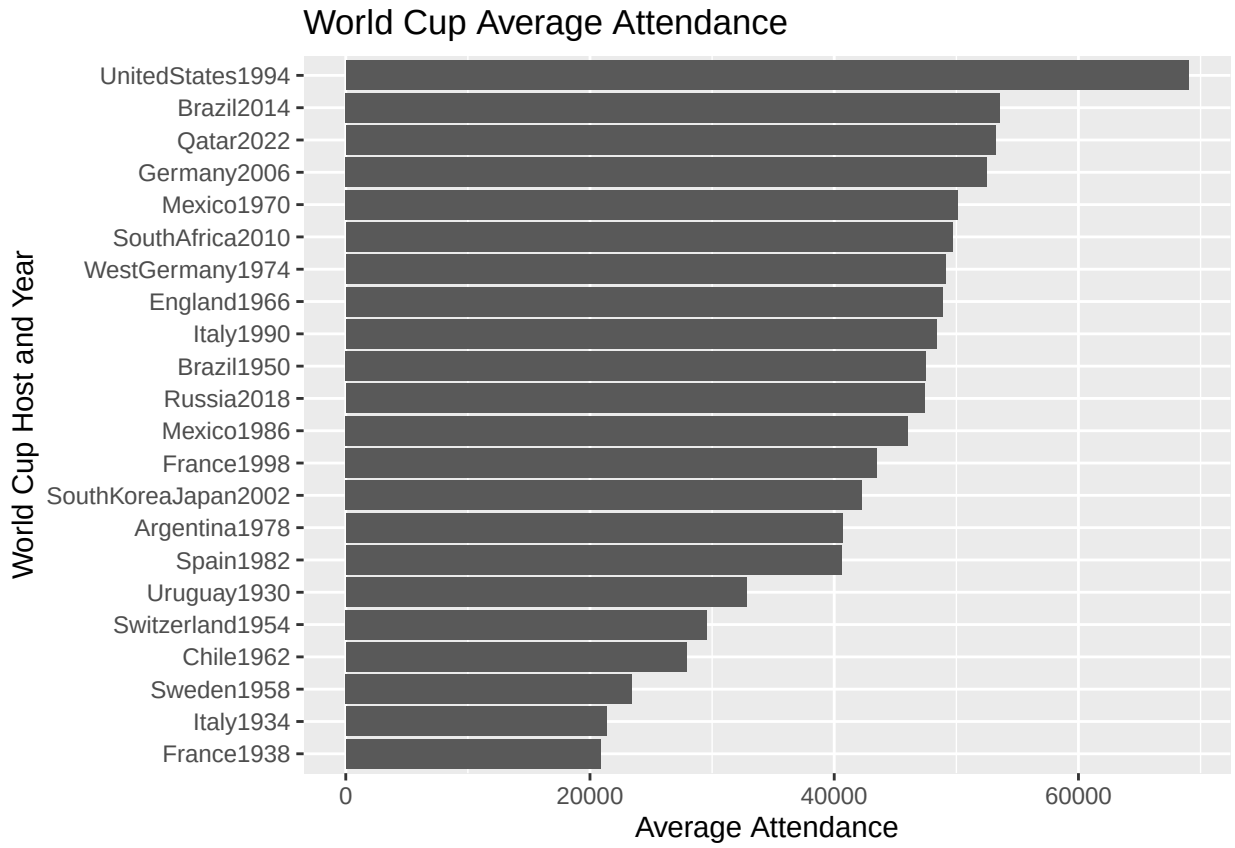
e. Display the `str()` structure of the data frame `World_Cup`. There should be 22 rows and 4 columns!

```
str(World_Cup)
```

```
## tibble [22 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Totalattendance  : num [1:22] 590549 363000 375700 1045246 768607 ...
##  $ Matches          : num [1:22] 18 17 18 22 26 35 32 32 32 38 ...
##  $ Averageattendance: num [1:22] 32808 21353 20872 47511 29562 ...
##  $ WorldCup         : chr [1:22] "Uruguay1930" "Italy1934" "France1938" "Brazil1950" ...
```

f. Create a column graph displaying `WorldCup` against the `Averageattendance`. Arrange the graph such that the bars are ordered by average attendance. Make sure the `WorldCup` identifiers are visible on the graph (i.e. you can read them). Clean up the axes such that they read `World Cup Host and Year` and `Average Attendance`.

```
# Reorder Data
World_Cup<-World_Cup%>%mutate(WorldCup=fct_reorder(WorldCup,Averageattendance))

P<-ggplot(data=World_Cup,aes(x=Averageattendance,y=WorldCup))+geom_col()+
  ggtitle("World Cup Average Attendance")+
  labs(x="Average Attendance",y="World Cup Host and Year")

P
```



Question 2 [20 points]

Considering the average attendance at World Cup matches got me thinking about world population. I was able to find an excel file from the United Nations tracking estimated populations for all countries that are part of the UN. This data is available as `World_Populations.xlsx` within the `Final Exam` assignment folder.

a. Load the data frame the `ESTIMATES` tab. Be sure to skip any uninformative lines.

```
myData<-read_excel('World_Population.xlsx',sheet='ESTIMATES',skip=16)
```

b. Using regular expressions and tidyverse commands, clean the data to include only population information from 1950 to 2020 for all countries. Remove all extra information regarding regions, subregions, income, etc. Retain only the `Country Name` and population estimates for years 1950 to 2020. Name this data.frame `WorldPopulation` and show the `head()` when finished.

```
WorldPopulation<-myData%>%filter(Type=="Country/Area")%>%
  select(-1,-2,-4,-5,-6,-7)%>%
  rename("Country"=1)



# Show the data
head(WorldPopulation)
```

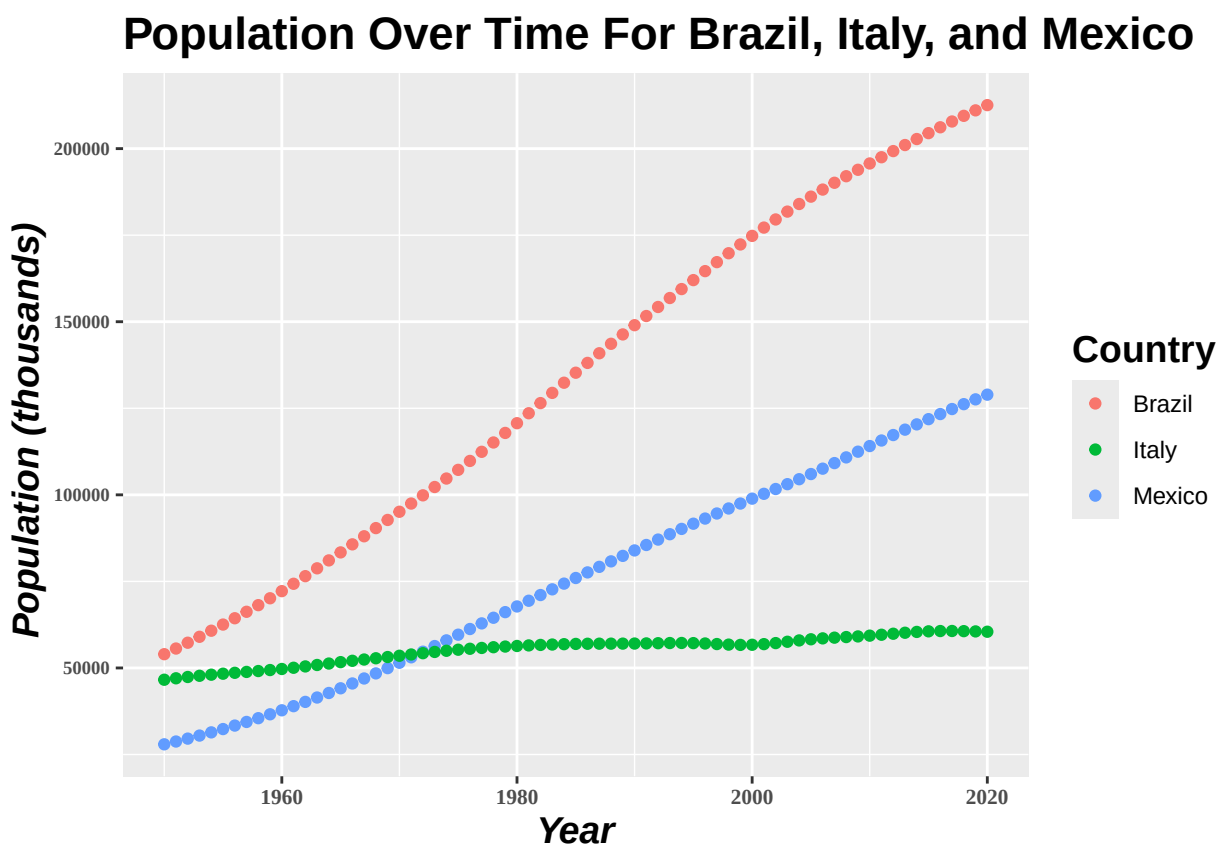
```
## # A tibble: 6 x 72
##   Country  `1950` `1951` `1952` `1953` `1954` `1955` `1956` `1957` `1958` `1959`
##   <chr>    <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
## 1 Burundi  2308.~ 2360.~ 2406.~ 2449.~ 2492.~ 2537.~ 2584.~ 2635.~ 2688.~ 2743.~
## 2 Comoros  159.4~ 163.1~ 166.5~ 169.7~ 172.8~ 175.9~ 178.9~ 181.99 185.0~ 188.0~
## 3 Djibouti 62     63.31~ 64.744 66.27~ 67.884 69.59~ 71.494 73.69~ 76.35~ 79.61~
## 4 Eritrea  822.3~ 835    849.2~ 864.8~ 881.7~ 899.7~ 918.8~ 939.0~ 960.5~ 983.3~
## 5 Ethiopia 18128~ 18466~ 18819~ 19184~ 19560~ 19947~ 20347~ 20764~ 21201~ 21661~
## 6 Kenya    6076.~ 6242.~ 6415.~ 6598.~ 6788.~ 6987.~ 7195.~ 7411.~ 7637.~ 7873.~
## # i 61 more variables: `1960` <chr>, `1961` <chr>, `1962` <chr>, `1963` <chr>,
## #   `1964` <chr>, `1965` <chr>, `1966` <chr>, `1967` <chr>, `1968` <chr>,
## #   `1969` <chr>, `1970` <chr>, `1971` <chr>, `1972` <chr>, `1973` <chr>,
## #   `1974` <chr>, `1975` <chr>, `1976` <chr>, `1977` <chr>, `1978` <chr>,
## #   `1979` <chr>, `1980` <chr>, `1981` <chr>, `1982` <chr>, `1983` <chr>,
## #   `1984` <chr>, `1985` <chr>, `1986` <chr>, `1987` <chr>, `1988` <chr>,
## #   `1989` <chr>, `1990` <chr>, `1991` <chr>, `1992` <chr>, `1993` <chr>, ...
```

c. Create a single panel graph displaying `Year` against `Population` for `Brazil`, `Mexico`, and `Italy`. Use different colors for the three countries. Properly label the axes.

```
myDataC<-WorldPopulation%>%
  filter(Country=="Brazil" | Country=="Mexico" | Country=="Italy")%>%
  pivot_longer(2:72,names_to = "Year",values_to="Population")%>%
  mutate(Year=as.numeric(Year))%>%
  mutate(Population=as.numeric(Population))

# Manipulating the table so that it is easier to graph




PC<-ggplot(data=myDataC,aes(x=Year,y=Population))+geom_point(aes(color=Country))+
  ggtitle("Population Over Time For Brazil, Italy, and Mexico")+
  labs(x="Year",y="Population (thousands) ")
```

d. Apply a theme of your choice to the graph in part **(c)**.

```
PD<-PC+theme(
  # Change x-axis font type
axis.text.x = element_text(size = 8, face = "bold", family = "serif"),
# Change y-axis font type
axis.text.y = element_text(size = 7, face = "bold", family = "serif"),
# Change x-axis title font type
axis.title.x = element_text(size = 14, face = "bold.italic"),
# Change y-axis title font type
axis.title.y = element_text(size = 14, face = "bold.italic"),
title=element_text(size=14,face="bold")
)

PD
```



Question 3 [20 points]

I want to be able to easily graph any of the UN countries given in the Excel file for Question 2. My preference would be to just enter a country name and obtain a graph of the population from 1950 to 2020.

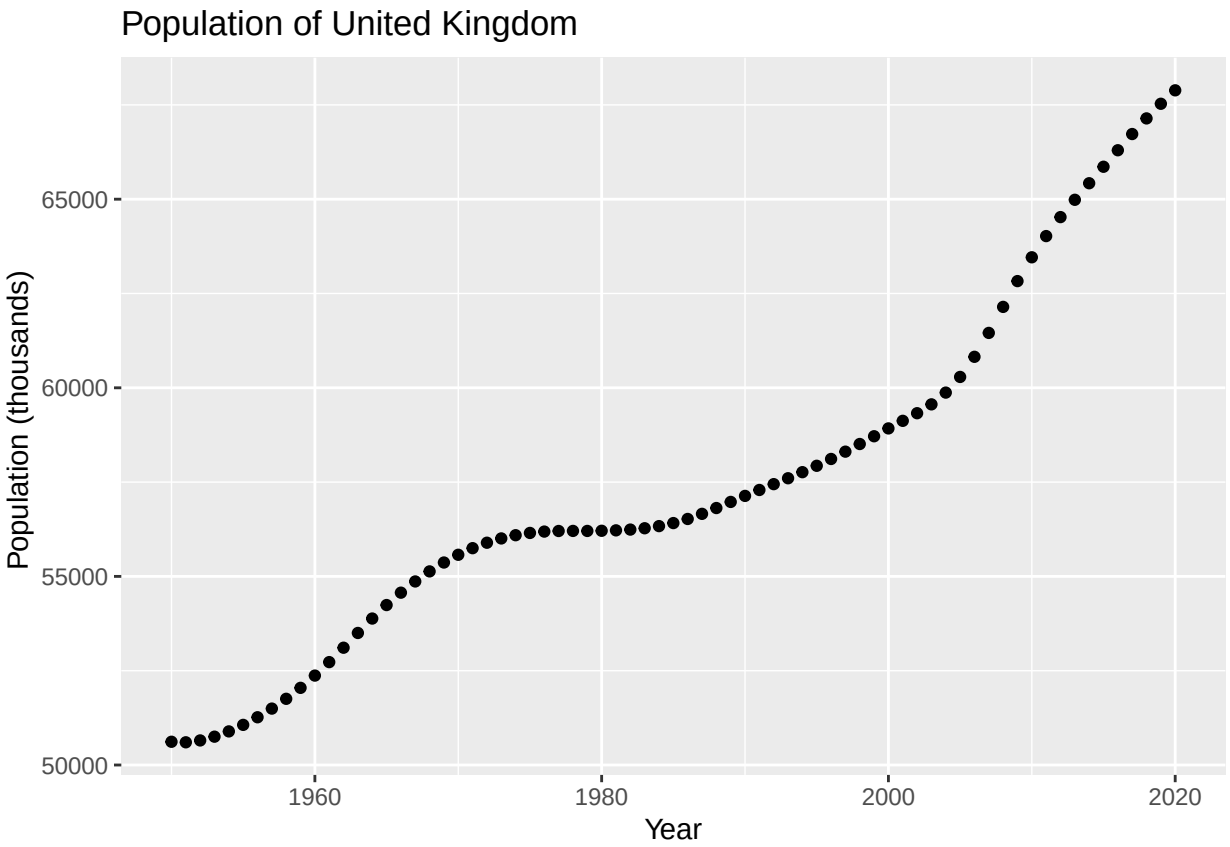
a. Produce a function that uses the `WorldPopulation` data.frame from Question 2 part (b) to generate a graph of any countries population over time. That is, `WorldPopulation` should NOT be an input variable. The function should only take as input a country name (as a string - such as `Italy`) and return the population against year graph for that country. The name of the country should be within the title of the graph and the axes should be properly labeled. Name this function `CountryPopulation`.

Hint: Wrap up what you did Question 2c into a function that returns an object that is a ggplot. Remove any options for color. Add an option for title that uses the input string. This should produce a black and white graph with the name of the country at the top.

```
CountryPopulation<- function(country){
  tempData<-WorldPopulation%>%filter(Country==country)%>%
  pivot_longer(2:72,names_to = "Year",values_to="Population")%>%
  mutate(Year=as.numeric(Year))%>%
  mutate(Population=as.numeric(Population))

# Manipulating the table so that it is easier to graph


title=paste("Population of",country)

Out_Plot<-ggplot(data=tempData,aes(x=Year,y=Population))+geom_point()+
  ggtitle(title)+
  labs(x="Year",y="Population (thousands) ")

return(Out_Plot)
}
```

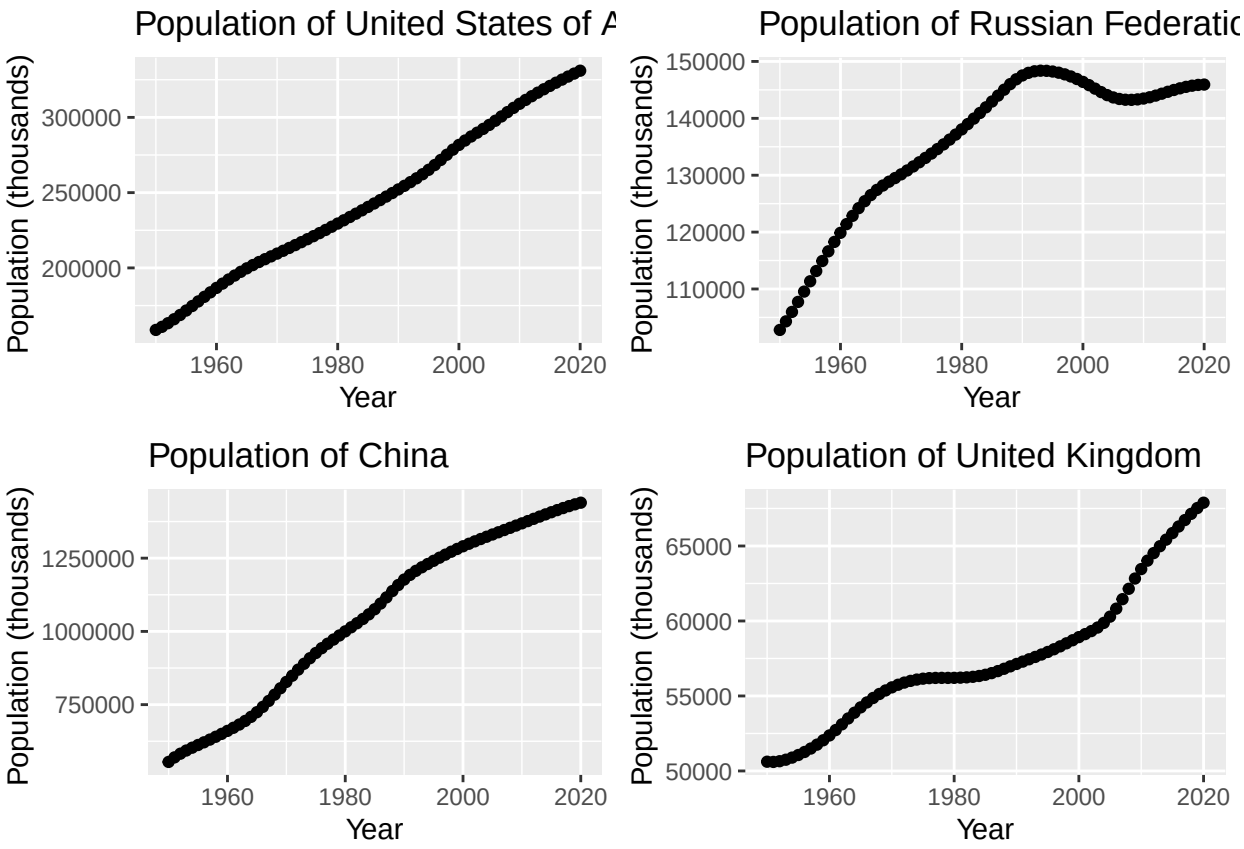
b. Using your function `CountryPopulation` produce graphs for `United States of America`, `Russian Federation`, `China`, and `United Kingdom`. Store these as objects to be used in part **(c)**. Display the graph for `United Kingdom`.

```
USA_graph<-CountryPopulation("United States of America")
Russia_graph<-CountryPopulation('Russian Federation')
China_graph<-CountryPopulation("China")
UK_graph<-CountryPopulation("United Kingdom")


UK_graph
```



c. Using the `cowplot` package combine the four graphs from part **(b)** into a single graph.

```
cowplot::plot_grid(USA_graph,Russia_graph,China_graph,UK_graph)
```



Question 4 [35 points]

To receive credit for the below work, provide the link to your GitHub package within your submission PDF.

We now have some really interesting World Cup and World Population data as well as a function that allows us to view any population graphs of UN countries. Let's package this up with some additional troubleshooting. Follow the steps below and ensure you upload the package to your GitHub account. I would recommend double checking this works in some way - you CANNOT ask a classroom peer to do this as we did for the R Package assignment.

a. Initialize a new package named `YourLastNameWorldPopulation`.

Check

b. Add the `World_Population.xlsx` file to the `data-raw` folder.

Check

c. Using your cleaning script from **Question 2b**, add the cleaned version of your `WorldPopulation` data to the package. Document the data set.

Check

d. Add your cleaned `World_Cup` data, with documentation, to your package.

Check

e. Add to your package the function `CountryPopulation`. Be sure to include a description for the documentation. Update the function such that if provided a country name that does not exist within your `World_Population` data, the function will return an error.

Check

f. Produce a unit test to the package to check if a country name entered is in the cleaned data file `WorldPopulation`. If the country is not present, then the function `CountryPopulation` should return an error.

Check

g. Compile your package and upload to your GitHub within the repository `YourLastNameWorldPopulation`.

h. As a solution to Question 4, provide the link to your GitHub package. The package should be able to install directly from GitHub to receive credit for this question. Your package should include the following items, with documentation, when finished: `WorldPopulation`, `World_Cup`, `CountryPopulation`.